

# SciDAC Software Infrastructure for Lattice Gauge Theory

Richard C. Brower  
Annual Progress Review  
JLab , May 14 , 2007

Code distribution see  
<http://www.usqcd.org/software.html>

# Software Committee

- Rich Brower (chair)      [brower@bu.edu](mailto:brower@bu.edu)
- Carleton DeTar              [detar@physics.utah.edu](mailto:detar@physics.utah.edu)
- Robert Edwards              [edwards@jlab.org](mailto:edwards@jlab.org)
- Don Holmgren                [djholm@fnal.gov](mailto:djholm@fnal.gov)
- Bob Mawhinney              [rdm@phys.columbia.edu](mailto:rdm@phys.columbia.edu)
- Chip Watson                 [watson@jlab.org](mailto:watson@jlab.org)
- Ying Zhang                  [yingz@renci.org](mailto:yingz@renci.org)

# Major Participants in SciDAC Project

Arizona	Doug Toussaint	MIT	Andrew Pochinsky
	Dru Renner		Joy Khoriaty
BU	Rich Brower *	North Carolina	Rob Fowler
	James Osborn		Ying Zhang *
	Mike Clark	JLab	Chip Watson *
BNL	Chulwoo Jung		Robert Edwards *
	Enno Scholz		Jie Chen
	Efstratios Efstathiadis		Balint Joo
Columbia	Bob Mawhinney *	IIT	Xien-He Sun
DePaul	Massimo DiPierro	Indiana	Steve Gottlieb
FNAL	Don Holmgren *		Subhasish Basak
	Jim Simone	Utah	Carleton DeTar *
	Jim Kowalkowski		Ludmila Levkova
	Amitoj Singh	Vanderbilt	Ted Bapty

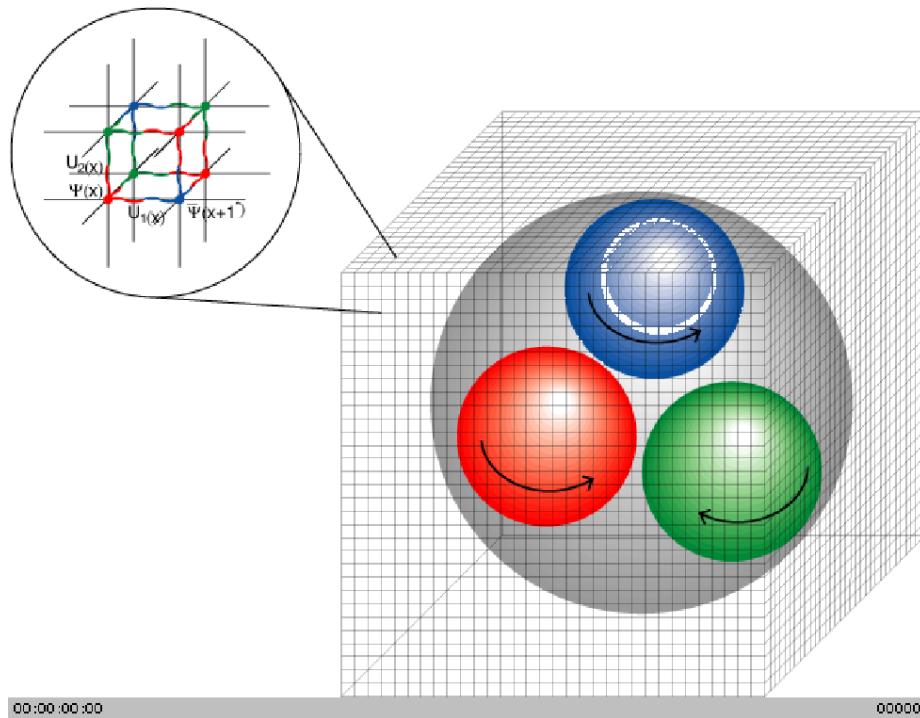
\* Software Committee: Participants funded in part by SciDAC grant

**QCD Software Infrastructure Goals:** *Create a unified software environment that will enable the US lattice community to achieve very high efficiency on diverse high performance hardware.*

**Requirements:**

- Build on 20 year investment in MILC/CPS/Chroma
- Optimize critical kernels for peak performance
- Minimize software effort to port to new platforms & to create new applications

# Solution for Lattice QCD



- (Perfect) Load Balancing:  
*Uniform periodic lattices & identical sublattices per processor.*
- (Complete) Latency hiding:  
*overlap computation /communications*
- Data Parallel: *operations on Small 3x3 Complex Matrices per link.*
- Critical kernels : *Dirac Solver, HMC forces, etc. ... 70%-90%*

Lattice Dirac operator:

$$[D\Psi]_\alpha^i(x) = \frac{1}{2a} \sum_\mu [U_\mu^{ij}(x) \gamma_\mu^{\alpha\beta} \Psi_\beta^j(x+\mu) - h.c] + \dots$$

# SciDAC-1 QCD API

Optimised for P4 and QCDOC

Level 3

Optimized Dirac Operators,  
Inverters

ILDG collab

Level 2

QDP (QCD Data Parallel)  
Lattice Wide Operations,  
Data shifts

QIO  
Binary/ XML  
Metadata Files

Level 1

QLA (QCD Linear Algebra)

QMP (QCD Message Passing)

Exists in C/C++

C/C++, implemented over MPI, native  
QCDOC, M-via GigE mesh

# Data Parallel QDP/C,C++ API

- Hides architecture and layout
- Operates on lattice fields across sites
- Linear algebra tailored for QCD
- Shifts and permutation maps across sites
- Reductions
- Subsets
- Entry/exit – attach to existing codes

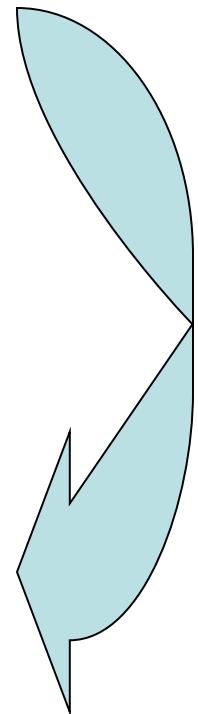
# Example of QDP++ Expression

- *Typical for Dirac Operator:*

$$\psi_\alpha^i(x) = U_\mu^{ij}(x)\chi_\alpha^j(x+\mu) + 2\phi_\alpha^i(x) \quad \forall i, \alpha, x \in \text{even}$$

- *QDP/C++ code:*

```
multi1d<LatticeColorMatrix> u[Nd] ;
LaticeDiracFermion psi, chi, phi ;
int mu;
psi[even] = shift[chi,mu] + 2 * phi ;
```



- *Use Portable Expression Template Engine (PETE)*

Temporaries eliminated, expressions optimized

Application Codes:

MILC

/ CPS

/ Chroma

/ RoleYourOwn

*PERI*

*TOPS*

# SciDAC-2 QCD API

Level\_4

## QCD Physics Toolbox

Shared Alg, Building Blocks, Visualization, Performance Tools

## Workflow

and Data Analysis tools

Level\_3

## QOP (Optimized in asm)

Dirac Operator, Inverters, Force etc

## Uniform User Env

Runtime, accounting, grid,

Level\_2

## QDP (QCD Data Parallel)

Lattice Wide Operations, Data shifts

## QIO

Binary / XML files & ILDG

Level\_1

## QLA

(QCD Linear Algebra)

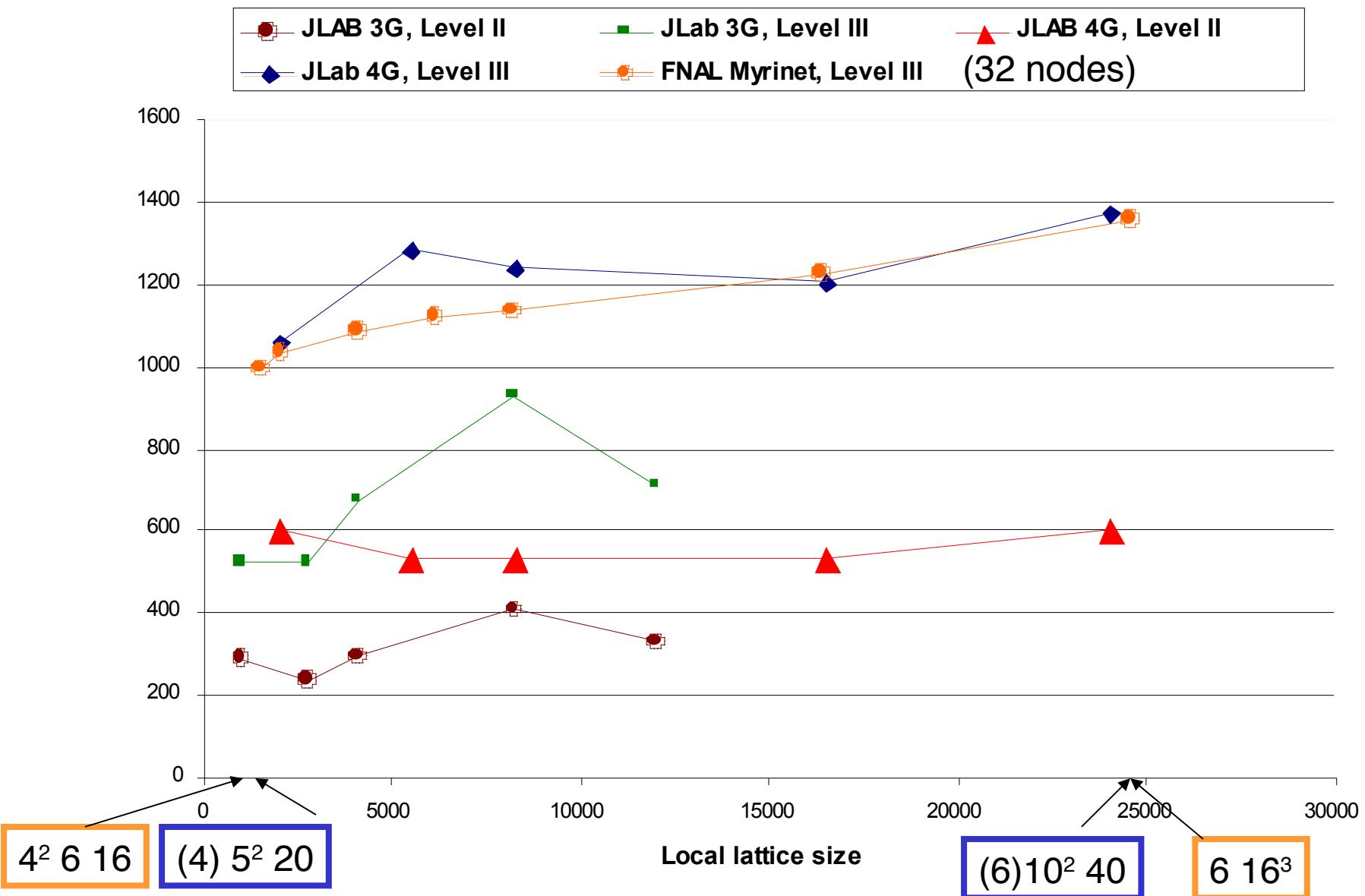
## QMP

(QCD Message Passing)

## QMC

(QCD Multi-core interface)

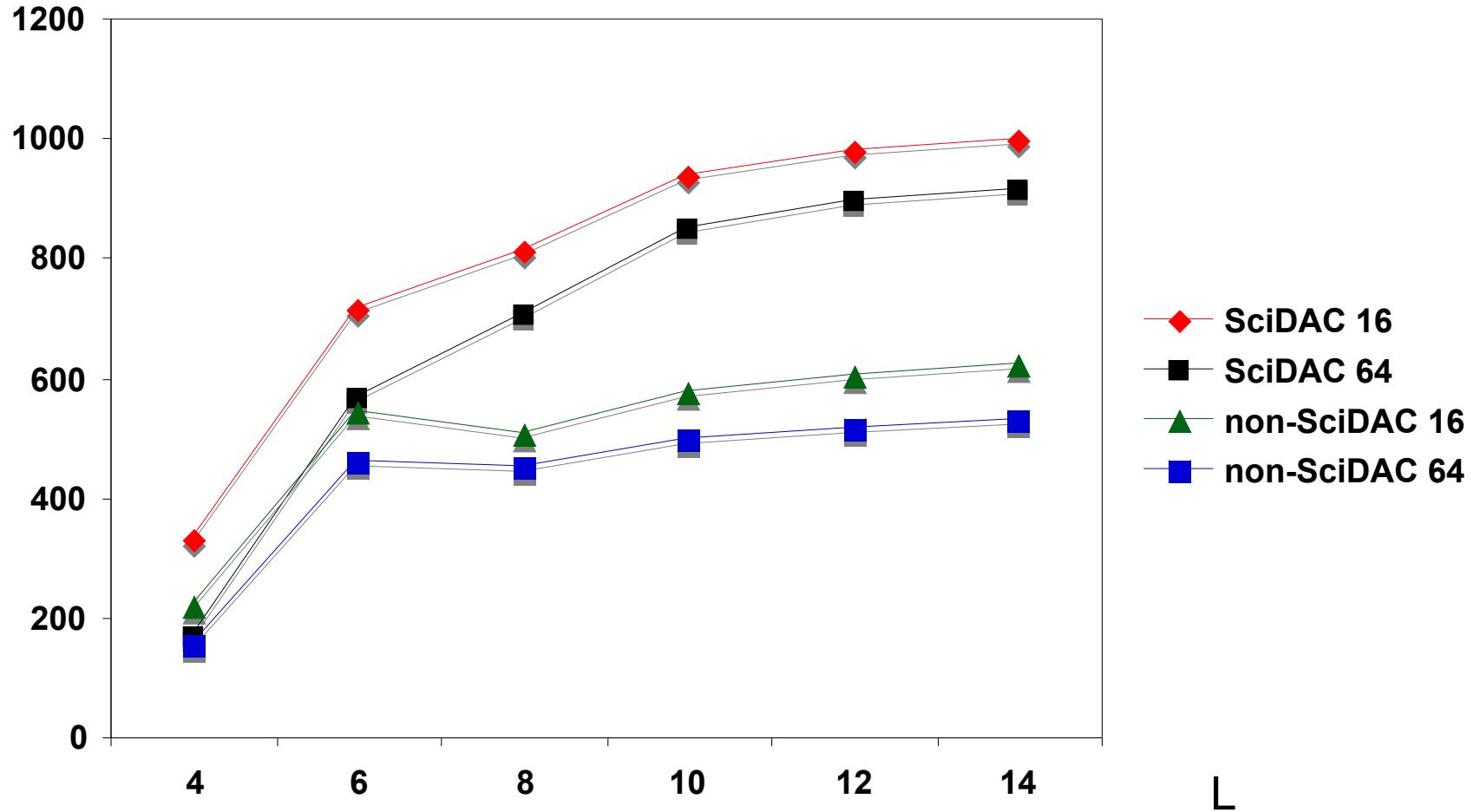
# Level 3 Domain Wall CG Inverter †



†  $L_s = 16$ , 4g is P4 2.8MHz, 800MHz FSB

# Asqtad Inverter on Kaon cluster @ FNAL

Mflop/s per core

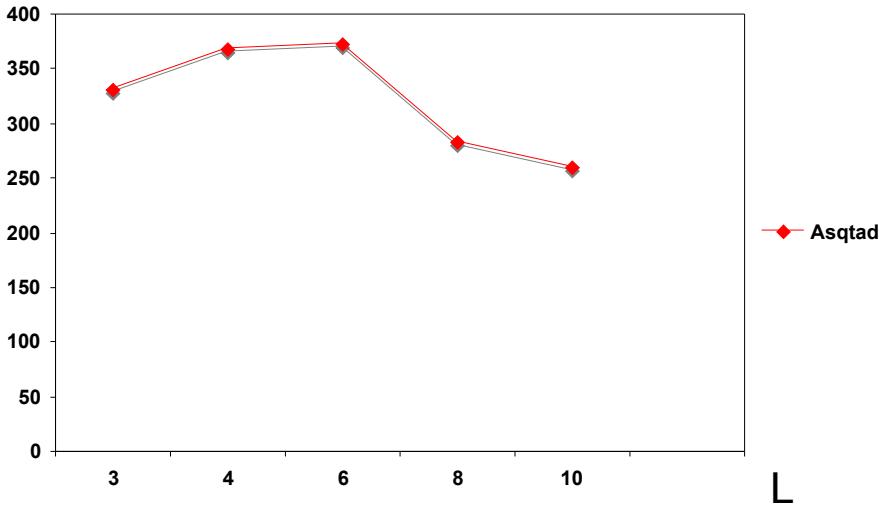


Comparison of MILC C code vs SciDAC/QDP on  $L^4$  sub-volumes for  
16 and 64 core partition of Kaon

# Level 3 on QCDOC

Dirac CG	Local Volume	Mflop/s
Domain Wall CG	$4^3 \times 8 \times 16$	268
Domain Wall CG	$6^3 \times 2 \times 8$	258
Asqtad CG	$4^4$	336
Asqtad CG	$8^4$	235
Asqtad multimass	$6^3 18$	175

Mflop/s



Asqtad CG on  $L^4$  subvolumes

## DW RHMC kernels

Routine	Mflop/s	% cpu time
CG Invert	268	29%
Multimass CG	210	55%
Fermion Force	28	13%
Gauge Force	26	1.5 %

$32^3 \times 64 \times 16$  with subvolumes  $4^3 \times 8 \times 16$

## Asqtad RHMC kernels

Routine	Mflop/s	% cpu time
CG Invert	156	46.6%
Fermion Force	223	16.6%
Gauge Force:	87	15.0%
Fat links	101	16.0 %
Naik links	73	6.2 %

$24^3 \times 32$  with subvolumes  $6^3 \times 18$

# Building on SciDAC-1

- ❖ Fuller use of API in application code.
  - q Integrate QDP into MILC & QMP into CPS
  - q Universal use of QIO, File Formats, QLA etc
  - q Level 3 Interface standards
- ❖ Common Runtime Environment
  - q File transfer, Batch scripts, Compile targets
  - q Practical 3 Laboratory “Metafacility”
- ❖ Porting API to INCITE Platforms
  - q BG/L & BG/P: QMP and QLA using XLC & Perl script
  - q Cray XT4 & Opteron, clusters

# New SciDAC-2 Goals

## ❖ Exploitation of Multi-core

- q Multi-core not Hertz is new paradigm
- q Plans for a QMC API (JLab & FNAL & PERC)

See SciDAC-2 kickoff workshop Oct27-28, 2006 <http://super.bu.edu/~brower/workshop>

## ❖ Tool Box -- shared algorithms / building blocks

- q RHMC, eigenvector solvers, etc
- q Visualization and Performance Analysis (DePaul & PERC)
- q Multi-scale Algorithms (QCD/TOPS Collaboration)

<http://www.yale.edu/QCDNA/>

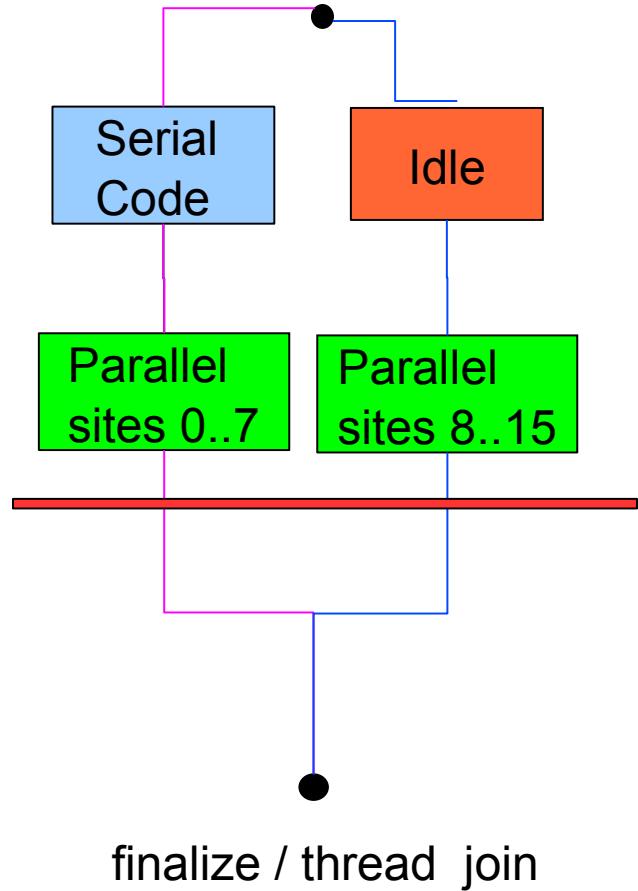
## ❖ Workflow and Cluster Reliability

- q Automated campaign to merge lattices, propagators and extract physics . (FNAL & Illinois Institute of Tech)
- q Cluster Reliability: (FNAL & Vanderbilt)

<http://lqcd.fnal.gov/workflow/WorkflowProject.html>

# QMC – QCD Multi-Threading

- General Evaluation
  - OpenMP vs. Explicit Thread library (Chen)
  - Explicit thread library can do better than OpenMP but OpenMP is compiler dependent
- Simple Threading API: QMC
  - based on older smp\_lib (Pochinsky)
  - use pthreads and investigate barrier synchronization algorithms
- Evaluate threads for SSE-Dslash
- Consider threaded version of QMP  
*( Fowler and Porterfield in RENCI)*



# Conclusions

- ❖ Progress has been made using a common QCD-API & libraries for Communication, Linear Algebra, I/O, optimized inverters etc.
- ❖ But full Implementation, Optimization, Documentation & Maintenance of shared codes is a continuing challenge.
- ❖ And there is much work to keep up with changing Hardware and Algorithms.

- ❖ Still NEW users (young and old) with no prior lattice experience have initiated new lattice QCD research using SciDAC software!
- ❖ The bottom line is PHYSICS is being well served.

Questions?